

Conception Orientée Objet

Projet: Tetris Attack (Version PC)

1 Le Jeu

Tetris Attack est un *puzzle-game* à 2 joueurs sorti en 1995 sur Super Nintendo. Ce jeu est une variante du bien connu Tetris (<https://fr.wikipedia.org/wiki/Tetris>) à 2 joueurs. Dans cette version, l'objectif est d'assembler des blocs entre eux afin de les faire disparaître avant que l'ensemble des blocs n'atteignent le haut de l'écran. Toutefois, ici, 2 joueurs s'affrontent en parallèle et peuvent à l'aide de différentes combinaisons envoyer de nouveaux blocs de briques à leur adversaire afin de précipiter sa perte. Ce jeu couple à la fois, la simplicité de Tetris à la dynamique nécessaire d'un bon Versus.

1.1 Le déroulement

Les 2 joueurs interagissent sur le même écran comme indiqué sur cette image¹ :



Le jeu est composé de n lignes de 6 blocs. Chaque bloc a pour caractéristique une couleur et un motif (coeur rouge, triangle bleu, étoile jaune, cercle vert, losange mauve ou exclamation gris).

Contrairement à Tetris, les lignes apparaissent par le bas de l'écran tout le long du jeu à une vitesse croissante au cours de la partie. Chaque ligne qui apparaît est générée aléatoirement.

L'objectif de chaque joueur est de réaliser des combinaisons de 3, 4 ou 5 blocs identiques en ligne ou en colonne.

Notons qu'il sera également possible de réaliser des combinaisons plus complexes comme par exemple de 7 blocs avec 5 blocs en ligne et 3 blocs en colonne en un seul coup. De plus, les combos doivent être pris en compte car ils font toute la complexité du jeu. Un exemple de combo est une combinaison de 3 blocs qui en disparaissant provoquera par la chute d'un autre bloc, une nouvelle combinaison. On parlera de combo X2.

Dans Tetris Attack, il est grandement conseillé de réaliser des combinaisons de plus de 3 blocs et de multiples combos car celle-ci provoque chez l'adversaire l'apparition de briques marons de différentes tailles comme répertoriés dans ce tableau :

Combinaison taille 4	brique taille 3
Combinaison taille 5	brique taille 5
Combo Xn	brique taille 6 à n-1 niveaux

Les briques qui apparaissent chez l'adversaire seront toujours positionnées le plus à gauche et le plus bas possible et seront prévenus 3 secondes avant leur apparition. Un joueur pourra transformer ces briques en blocs si il effectue une combinaison qui est connectée à ces briques. Toutefois, si une brique possède plusieurs niveaux, seul le niveau le plus inférieur sera transformé et une nouvelle combinaison sera nécessaire pour transformer le niveau suivant (et ainsi de suite).

Un joueur perd la partie si l'un de ses blocs (ou briques) reste plus de 2 secondes sur la 12ème ligne du jeu.

Bien que je vous ai décrit de manière la plus précise possible le jeu, cette vidéo <https://www.youtube.com/watch?v=TQdrINpVtNU> vous permettra de réaliser le déroulement d'un jeu. De nombreuses autres vidéos sont également disponibles sur internet.

¹ Le design demandé pour ce projet doit être proche de celui-ci. Par contre, il n'est pas demandé de reprendre l'ambiance "Nintendo" et les différentes animations qui se trouvent dans le jeu d'origine.

1.2 Les actions du joueur

Les actions du joueur sont simples.

Chaque joueur possède un curseur horizontal mobile sur sa partie lui permettant de permuter 2 blocs du jeu. Il peut le bouger autant qu'il le souhaite sur l'ensemble de son demi-écran. Il est également possible pour un joueur d'augmenter son demi-écran d'une ligne de blocs à sa guise, afin de lui permettre de créer de nouvelles combinaisons pour envoyer plus de briques à son adversaire.

1.3 L'état initial

Au début de chaque partie, chaque joueur démarre avec 6 lignes composées aléatoirement de différents blocs et aucune combinaison de cases n'est présente initialement. On laissera 3 secondes en début de partie à chaque joueur avant de démarrer les mouvements et la montée du jeu, ils seront indiquées par les mot-clé "Ready", "Steady", "Go". Initialement, le curseur de chaque joueur se trouve sur la 3ème ligne et le 3ème bloc.

1.4 Le score

Ce jeu présente aussi la faculté d'attribuer un score à chaque combinaison réalisée. L'attribution des scores doit être réalisée comme dans le tableau suivant :

Combinaison taille n	10 *(n-2)
Combo Xn	Somme des combinaisons * n

1.5 Quelques liens...

3 liens pour vous permettre d'en savoir plus sur Tetris Attack :

- https://en.wikipedia.org/wiki/Tetris_Attack
- http://strategywiki.org/wiki/Tetris_Attack
- http://www.mariowiki.com/Tetris_Attack

Un lien pour vous permettre de récupérer les images dont vous aurez besoin :

- <http://www.sprites-resource.com/snes/tetrisattack/>

Tester le jeu par vous-même en ligne sur navigateur :

- <http://game-oldies.com/play-online/tetris-attack-nintendo-super-nes#>

2 Votre mission...

Votre mission est la suivante :

2.1 Un *Git Repository*

Quand vous travaillez sur un projet de cette taille, il vous sera très souvent demandé de travailler via un *repository* en ligne. C'est pourquoi pour ce projet, je vous impose l'utilisation de Git². Si cet outil est nouveau pour vous, voici un bon tutoriel pour l'utiliser facilement sous Eclipse :

- <https://openclassrooms.com/courses/egit-quand-git-s-invite-dans-eclipse>

L'université propose une *forge* via *GitLab* à cette adresse <https://forge.univ-artois.fr/> vous permettant de créer votre projet sur laquelle travaillez.

Vous pouvez trouver les comptes de vos professeurs, veuillez nous ajouter afin que nous puissions observer l'avancement de vos projets respectifs.

2.2 Diagramme UML

La première étape de votre projet est d'établir un diagramme UML de votre projet via le logiciel de votre choix. Vous pouvez par exemple utiliser <http://plantuml.com/> ou <http://www.umldesigner.org/>.

Au cours de ce projet, le Modèle Modèle-Vue-Contrôleur (qui sera traité en cours) est de rigueur.

²Attention à bien gérer votre *repository* pour éviter les conflits et donc les pertes de code !

2.3 Menu

Votre jeu devra commencer par une première vue amenant sur un menu. Ce menu commencera par indiquer le titre du jeu (que je vous laisse modifier à votre guise) puis quelques sous-menus imposés :

- VS : Ce sous-menu permet de lancer le jeu rapidement. Une fois cliqué, nous devons directement arriver à une partie générée aléatoirement.
- Aide : Ce sous-menu indique les commandes à l'utilisateur et une description très rapide de votre jeu. Toutefois, la souris (ou les flèches du clavier et la touche "entrée") sont obligatoires pour se diriger dans le menu.
- Credits : Ce sous-menu doit me permettre d'atteindre une page qui indique votre nom et prénom, votre groupe de TP, l'ensemble des problèmes connus de votre logiciel et l'ensemble des fonctionnalités que vous avez ajouté à votre logiciel qui fonctionnent. Vous pouvez ajouter tout ce qu'ils vous semblent utiles pour vous noter et apprécier votre projet.
- Quitter : Qui permettra de quitter l'application

2.4 Design

Le design de votre projet doit respecter globalement celui de Nintendo. Pour vous aider, je vous met à disposition sur un *repository git* des images du jeu (de mon design personnel...) qui peuvent vous faciliter la tâche : <https://epiette@bitbucket.org/epiette/projet-coo-l3-2016.git>

Toutefois, vous êtes libres de le modifier si vous en sentez la nécessité pour simplifier le Gameplay ou l'appréciation de votre projet. Chaque modification devra être présente dans vos crédits.

Pour vous aider à commencer à "jouer" avec Swing, je vous conseille ce site <http://www.jmdoudoux.fr/java/dej/chap-swing.htm> qui vous rappellera les bases.

2.5 Interactions Homme Machine

Evidemment seul un clavier devra être utilisé pour que les 2 joueurs jouent. Soyez donc malin sur les touches allouées afin de simplifier le *GamePlay*. Dans le cas optimal (et si vous avez assez de temps et de courage), la modification des commandes devra être disponibles via le menu dans un nouvel onglet du menu et stocké dans un fichier XML de configuration associé à votre projet.

2.6 Structure de données

Vous êtes complètement libre sur la structure de données. Cependant, prenez votre temps en début de projet pour en choisir une qui permettra de réaliser l'ensemble des fonctionnalités et qui ne vous posera aucune limitation ou difficulté, vous imposant ainsi la refonte de l'ensemble de votre projet. SOYEZ VIGILANT !

2.7 Une I.A.

La partie qui vous apportera le plus de points est de réaliser une Intelligence Artificielle (I.A.) qui jouera contre vous ou contre elle-même. Cette I.A. pourra comporter différents niveaux de difficultés et s'adapter au joueur afin de rendre le jeu le plus amusant possible. Dans le mode le plus difficile, votre I.A. devra être imbattable par l'humain. Si vous avez besoin de conseils pour cette partie, n'hésitez pas à m'interpeller lors des travaux pratiques ou par mail.

2.8 Changer les commandes

Permettez à votre utilisateur de modifier les commandes du jeu et de les sauvegarder dans un fichier XML afin de les retrouver à la prochaine partie.

2.9 Montons le son !

Intégrez un peu de musique à votre application. Pour cela, je vous impose l'utilisation d'une API qui est devenu culte dans le monde Java : JavaSound <http://www.oracle.com/technetwork/java/index-139508.html>. A vous de vous renseigner et d'utiliser cette API pour ajouter un son d'ambiance à votre jeu. Cependant, l'une des difficultés sera de permettre à votre application d'adapter le fond sonore selon les interactions des utilisateurs sans provoquer de ralentissement.

2.10 Envie d'une pause

Ajoutez une commande permettant de mettre le jeu en pause. Si on réappuie sur cette commande, le jeu reprend comme avant la pause.

2.11 Un petit dernier pour la route...

J'ai volontairement oublier une partie du jeu afin de vous simplifier la tâche. Celle-ci concerne les blocs "points d'exclamation", on les appelle plus couramment bloc gris. Ces blocs sont particuliers et ne sont pas soumis au même règle comme l'indique le tableau suivant :

Combinaison taille 3	brique taille 5
Combinaison taille 4	brique taille 6
Combinaison taille 5	brique taille 6 à 2 niveaux
Combo Xn	brique taille 6 à n niveaux

Ce type de bloc doit apparaître beaucoup moins souvent dans le jeu car il apporte un gros avantage au joueur. Ainsi leur apparition doit suivre une distribution aléatoire identique pour les 2 joueurs.

De plus, les briques grises réalisées par ces blocs doivent être détruits indépendamment des autres briques par une combinaison en contact avec eux.

2.12 Exécution

Votre projet devra s'exécuter par une simple ligne de commande que voici : `java -jar Nomduprojet.jar`

3 Notation

Votre projet sera noté sur plusieurs critères :

- Respect des consignes (UML, MVC, Git, etc...)
- Bon fonctionnement du jeu (menu, interactions, règles)
- Développement clair et commenté (une javadoc sera appréciée).³
- Intérêt et motivation pour le projet (n'hésitez pas à envoyer des mails et à poser des questions, voir même à faire des propositions et des versions préliminaires)
- Plus votre version sera proche de celle de Nintendo et plus elle sera facile d'utilisation de part un bon Gameplay, plus votre note sera bonne.
- L'ensemble des informations que vous aurez indiqués dans vos crédits et qui seront utiles pour la notation. Par exemple, les technologies utilisées, les sites et tutoriels suivis (et les éventuels remerciements ☺).

4 C'est parti !

Je vous souhaite à tous BON COURAGE !

Je vous rappelle nos mails si vous avez des questions :

- epiette@cril.fr
- vesic@cril.fr

J'attends vos questions ☺ (et problèmes, j'en suis certain)

Veuillez envoyer l'ensemble de votre projet par email à vesic@cril.fr (G1) ou à epiette@cril.fr (G2) entre le 10 avril 2016 à 8h et le 15 avril 2016 à 23h59 CET. N'oubliez pas le .jar.

Soutenance : le 22 avril 2016 - pendant le TP (présence obligatoire).

³Bien sûr, vous faites ici de l'objet et non de la programmation impérative, l'utilisation de l'ensemble de votre cours sera nécessaire